ARMY RESEARCH LABORATORY

Instance-Based Classification of Noisy Infrared Spectra

by Robert P. Winkler and Timothy C. Gregory

ARL-TR-1211

January 1997

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1197

ARL-TR-1211 **January 1997**

Instance-Based Classification of Noisy Infrared Spectra

Robert P. Winkler and Timothy C. Gregory
Information Science & Technology Directorate

Abstract

Successful systems for classification of real-world data must be tolerant of noise—that is, distortions introduced into the system's model of the real-world domain. Most classification systems are trained on a set of exemplars to identify features of each category and then tested on previously unseen instances. In an instance-based classification system using *k*-nearest neighbor (*k*-NN), the training phase is reduced to storing one or more exemplars for each category. During testing, a distance metric is applied to the features of the new instance to determine the *k* closest exemplars. A voting scheme assigns the category of the modal average to the testing instance. Unlike other methods, *k*-NN does not try to distinguish between "relevant" and "irrelevant" features. Nonetheless, *k*-NN has been shown to asymptotically approach optimal Bayesian accuracy.

This report presents the results of applying *k*-NN to the problem of classifying chemical agents from noisy infrared absorption spectra (from a suite of chemical agents used elsewhere in the literature). Straightforward nearest-neighbor approaches without editing appear to be tolerant of random noise when the amounts of noise in the training and testing sets are relatively close. Performance of *k*-NN versus 1-NN approaches can be improved if the training sets are edited so as to exclude degenerate outliers and redundant positive instances.

Contents

Figures

Tables

1. Introduction

The problem of noise or uncertainty pervades all natural classification domains. Noise is a distortion of the model contained in the classification system (as defined by the system's representation and bias and as indicated by its classification performance) from the observer's or experimenter's reality. Noise in a classification system can be of different types, arise from different sources, and be introduced at different points. Successful classification of real-world, "field" data must be tolerant of all these types of noise. In this report, we explore the effects of adding normal random noise on the classification of chemical agents from their infrared absorption spectra. In particular, we examine the classification performance of an instance-based classification technique known as *k*-nearest neighbor (*k*-NN) [1].

The data available for testing consist of 71 spectra of various chemical agents. The data were collected at a signal-to-noise ratio too high to be reproduced outside the laboratory environment. (Field data will have much lower signal-to-noise ratios.) Thus, the library spectra available for classification purposes are "prototypical." Of particular interest is how accurate these prototypical data are for classifying noisy data. To investigate this point, we added normal random noise to each spectrum to create noisy data sets. We then tested the classification system by training on the clean data and testing on the noisy, training on the noisy and testing on the clean, and finally by both training and testing on the noisy data.*

2. The Problem of Noise

Different types of noise include "white noise," "colored noise," and "clutter." Gaussian or "white" noise is uncorrelated with any specific values. "Colored noise" is correlated to specific values contained in or presented to the classification system. "Clutter" results from environmental conditions outside experimental control. Noise may exist in a number of different components within the classification system. It may be inherent in the classification system (probabilistic Bayesian belief networks or probabilistic search algorithms, for example). It may exist in the limitations, specifications, sensitivities, faults, etc, of the system's sensors providing the input. Noise may exist in the inputs themselves (because of variabilities unaccounted for in the environment, incomplete information—a specific feature not currently available, for example—or incorrect or distorted feature values). It may be introduced during the processing or presentation of the inputs (by probabilistic algorithms, for example). Finally, noise may even exist as the misclassification (either malicious or accidental) of the exemplars used for training. Any natural domain will inevitably contain all these types of noise to varying degrees. Any useful classification system will have to be tolerant of some level of noise.

**The permutation of clean training and clean testing was not feasible to this problem domain, since this approach would require multiple samples of every category.*

If we define noise as a distortion of the reality being modeled, then the better the classification system represents the domain being modeled, the lower the intrinsic noise. For all practical (as opposed to philosophical) purposes, the noise that exists in the inputs to the system and that introduced by the system's sensors are indistinguishable—they both result in distortions or perturbations to the values of the features or attributes presented to the system. Noise from misclassification of the training exemplars is discounted by some researchers as "insufficient to characterize practical situations" [2]. In fact, if we consider the final classification as just another attribute or feature of the input, noise of this type is just as prevalent as the other types. So ultimately, the source and nature of the noise is irrelevant to the classification system. Noise intrinsic to the classification system, regardless of its source or nature, is ultimately what is to be minimized.

Researchers in machine learning have long recognized the need for noise tolerance in classification systems [3–10]. Most of these researchers understand the importance of noise tolerance but fail to consider it in their own research. Connectionist-based approaches using artificial neural networks with back propagation have been shown to be tolerant of noise [8,11]. Rauss [11] has shown that in this problem domain, for the classification of these chemical agents, neural networks are tolerant of noise levels as high as 200 percent.* Researchers in instance-based and decision-tree methods claim that in these approaches (unlike connectionist-based approaches) additional modifications to the algorithms are necessary to handle noisy or uncertain information [4,5]. Some instance-based researchers even propose that noisy instances (i.e., "outliers") should be discarded and removed from consideration [12]. These researchers must have already examined the performance of these systems in the presence of noisy or uncertain information and found it to be inadequate. These results, however, are not typically presented. This report quantifies the effects of different levels of normal random noise on the accuracy of *k*-NN in this infrared spectral identification problem.

3. Problem Domain

The domain used for this test consisted of a library of 71 representative absorption spectra of various chemical agents. The library data were collected in a controlled laboratory environment at very high signal-to-noise ratios. The spectra thus represent "clean," "noise-free" prototypes for each agent. The spectra are idealizations; "live" samples from an actual sensor in an uncontrolled environment that happen to match these prototypes are considered merely coincidental and not expected. Field data for this domain are subject to numerous uncontrollable influences, including sensor limitations (assumed to be normally distributed) and environmental

**Since Rauss's method of introducing noise into the system [11] is different from ours in these experiments, we cannot directly compare his results with ours. His paper gives details on his approach.*

conditions (e.g., interference from other chemical agents). All these factors complicate the problem of classifying field data on the basis of the library data.

The sensor used to collect these data was a Fourier transform infrared (FTIR) spectrometer. Each spectrum consists of 571 features corresponding to the infrared absorption rate of the agent at wave numbers from 7.4 to 12.5 μm—the size of the FTIR window. Each spectrum was then normalized between –1 and 1. Rauss [13] has shown that several traditional curve-matching techniques (absolute value, least-squares fit, Euclidean distance, first derivative least-squares fit, and least-squares distance) have been unsuccessful in classifying the agents with various levels of noise. But Rauss was examining classification accuracy of the noisy signals by comparison to only the clean library spectra. In this report, we empirically quantify the performance of the *k*-NN algorithm (using Euclidean distance) when trained and tested on spectra with various levels of noise added.

4. Specification and Approach

An input instance to the system is specified as an ordered pair consisting of a 571-dimension feature vector and an integer value denoting the correct target classification for the feature vector:

$$I = (\langle f_1, f_2, \ldots, f_{571} \rangle, c_t) \tag{1}$$

where $f_i \in [0.0, 1.0]$ is the normalized value of the spectra at the i^{th} wavenumber, and $c_t \in [1, 71]$ represents the correct target class. During training, the correct classification is made available to the classification system. During testing, we compare the correct classification to the system's classification to determine classification accuracy.

The classification approach tested here is an instance-based method known as *k*-nearest neighbor [13]. The *k*-NN method has been shown to approach Bayesian classification as *k* increases [1]. We implemented the *k*-NN search algorithm using *k-d* trees [14]. The *k-d* tree algorithm superimposes a binary tree structure over the 571 features. This provides for retrieval of the *m* closest training examples in expected $\log_2(N)$ time (where *N* is the number of training examples). The mechanism for balancing the tree lies in ranking the selection of which feature to use as the discriminant for each node. The binary tree structure is balanced by a branch-and-bound strategy, which selects that feature for branching which exhibits the greatest variance over all the remaining instances (i.e., instances that fall below this node in the tree). Thus, the feature that shows the greatest variance over all training instances becomes the discriminant feature for the tree's root node, and so on. The partition value for each node is the median value of the node's discriminant feature. The search for the *m* training instances closest to a testing instance then proceeds down the tree. A priority queue of the *m* closest records encountered so far is maintained as the search progresses. The termination test is crucial for expected $\log_2(N)$ performance when *m* is greater than 1. Two 571-dimensional tuples are maintained, representing the lower and upper bounds for the features visited so far in the current subtree. A "ball" (hypersphere) around the testing instance is defined with radius equal to the distance between the testing instance and the m^{th} closest training instance encountered so far. The search can be terminated whenever the ball lies entirely within either the upper or lower bound. Backtracking through the tree is necessary only if the ball around the testing instance intersects the bound for the subtree opposite the current one (i.e., intersection with the lower bound if the ball is centered in the right subtree or intersection with the upper bound if the ball is centered in the left subtree).

5. Experimental Design

As mentioned earlier, sources of noise in this domain result from sensor limitations, environmental conditions, and interference from the presence of other chemical agents. Given that the only data available were clean spectra, the problem faced here is to add simulated noise to the clean library data. In this report we simulate noise by adding various levels of white noise to the original spectrum; the noise is distributed around a mean of 0.0 with variance equal to the level of noise times the variance* of the original spectrum. We then renormalized the resultant data between 0 and 1 by finding the minimum and maximum feature values of the new noisy spectrum. If the minimum value is less than 0, the difference between 0 and the minimum value is added to all the feature values. Each feature value is then divided by the maximum value (if the maximum value exceeds 1). We also normalized the original, unnormalized spectra (without added noise) between 0 and 1 by this same method to create the clean data set. Noise was added according to this definition in levels of 5, 10, 25, 50, 75, 100, 150, and 200 percent 10 times to each library spectrum, resulting in noisy data sets of 710 spectra at each of the noise levels. Figure 1 shows a sample spectrum with the various levels of noise added.

5.1 Experiment 1: Clean Training/Noisy Testing

For the clean training/noisy testing experiment, the 71 library spectra were used for training purposes, and the noisy data sets (a total of 8 consisting of 710 noisy spectra each) were used for testing. No cross-validation method could be used in this experiment, since there was only a single training instance for each category. For the same reason, only the closest matching library spectra were considered (1-nearest neighbor). Table 1 and figure 2 summarize the results of this experiment.

**Initially, the level of noise used was a percentage of the standard deviation. But it was noted that adding 10-percent noise to the standard deviation was equivalent to adding only 1-percent noise to the variance. The variance was selected because the resultant amount of noise was greater (until the noise level reached 100 percent).*

Figure 1. Sample spectrum with various levels of noise.

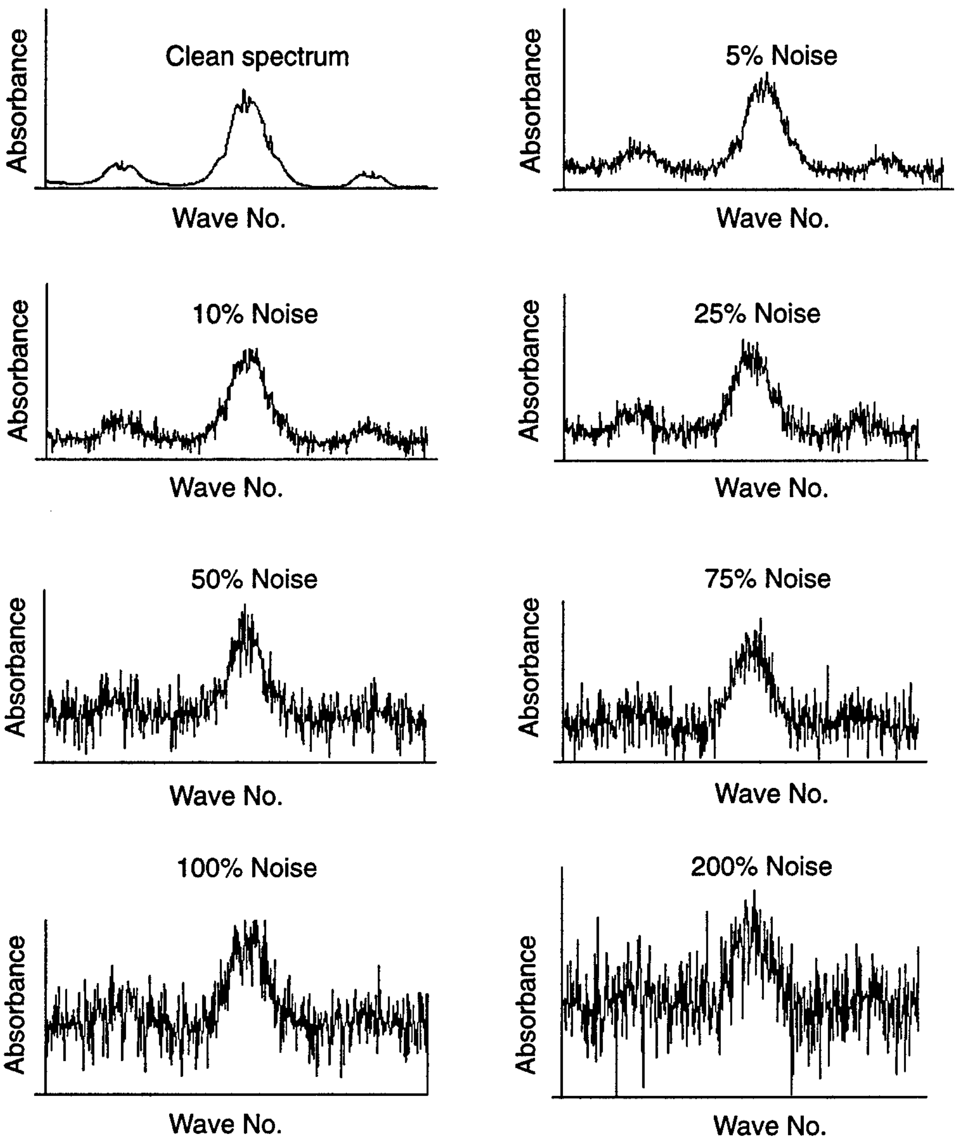


Table 1. Clean training (71 samples)/noisy testing (710 samples).

Testing noise level (%)	% correct classification (1-NN)
5	95.2
10	93.7
25	75.5
50	39.3
75	27.3
100	17.2
150	10.1
200	5.8

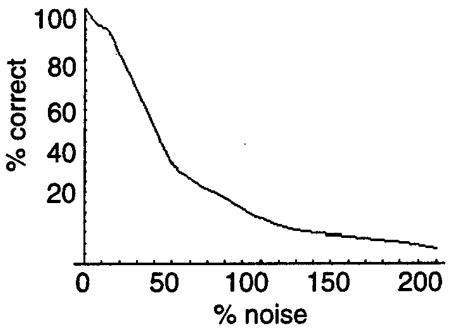


Figure 2. Clean training/noisy testing classification accuracy.

5.2 Experiment 2: Noisy Training/Clean Testing

For the noisy training/clean testing experiment, the 71 clean library spectra were used for testing, while the eight noisy data sets were used for training. No cross-validation methods applied here, for the same reasons as in the previous experiment. Unlike experiment 1, however, more than 1-nearest neighbor could be used, since the training set consisted of multiple examples (10) for each category. This experiment used 1-NN, 3-NN, 7-NN, and 17-NN. For *k*-NN where *k* was greater than 1, we used a voting scheme or modal average to determine the system's classification. (That is, for 3-NN, if the second and third closest training instances were of category A, while the first closest was of category B, the system would classify the test instance as category A.) Ties are biased in favor of the closest training instance. The results of this experiment are summarized in table 2 and figure 3.

5.3 Experiment 3: Noisy Training/Noisy Testing

For the noisy training/noisy testing experiment, the 710 library spectra at each noise level were randomly divided (assuming a uniform distribution) into training and testing data sets. The training data sets contained 639 spectra, while the test sets had the remaining 71 spectra. This division was repeated 10 times to each noisy data set, and tenfold cross-validation was used to calculate the average classification accuracies. Each training data set was tested against every test set. The testing used 1-NN and 7-NN. Tables 3 and 4 and figure 4 summarize the results of this experiment.

Table 2. Noisy training (710 samples)/clean testing (71 samples).

Training noise level (%)	% correct classification			
	1-NN	3-NN	7-NN	17-NN
5	95.8	95.8	93.0	90.1
10	84.5	80.3	71.8	45.1
25	9.9	8.5	7.0	9.9
50	1.4	1.4	2.8	2.8
75	4.2	4.2	2.8	4.2
100	4.2	4.2	2.8	1.4
150	4.2	2.8	2.8	2.8
200	2.8	2.8	2.8	1.4

Figure 3. Noisy training/clean testing classification accuracy (1-NN).

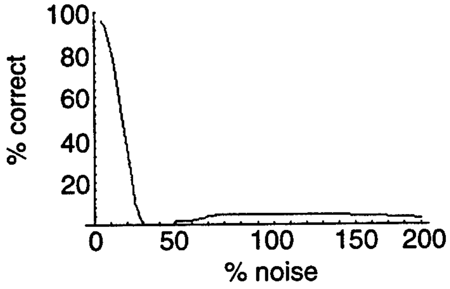


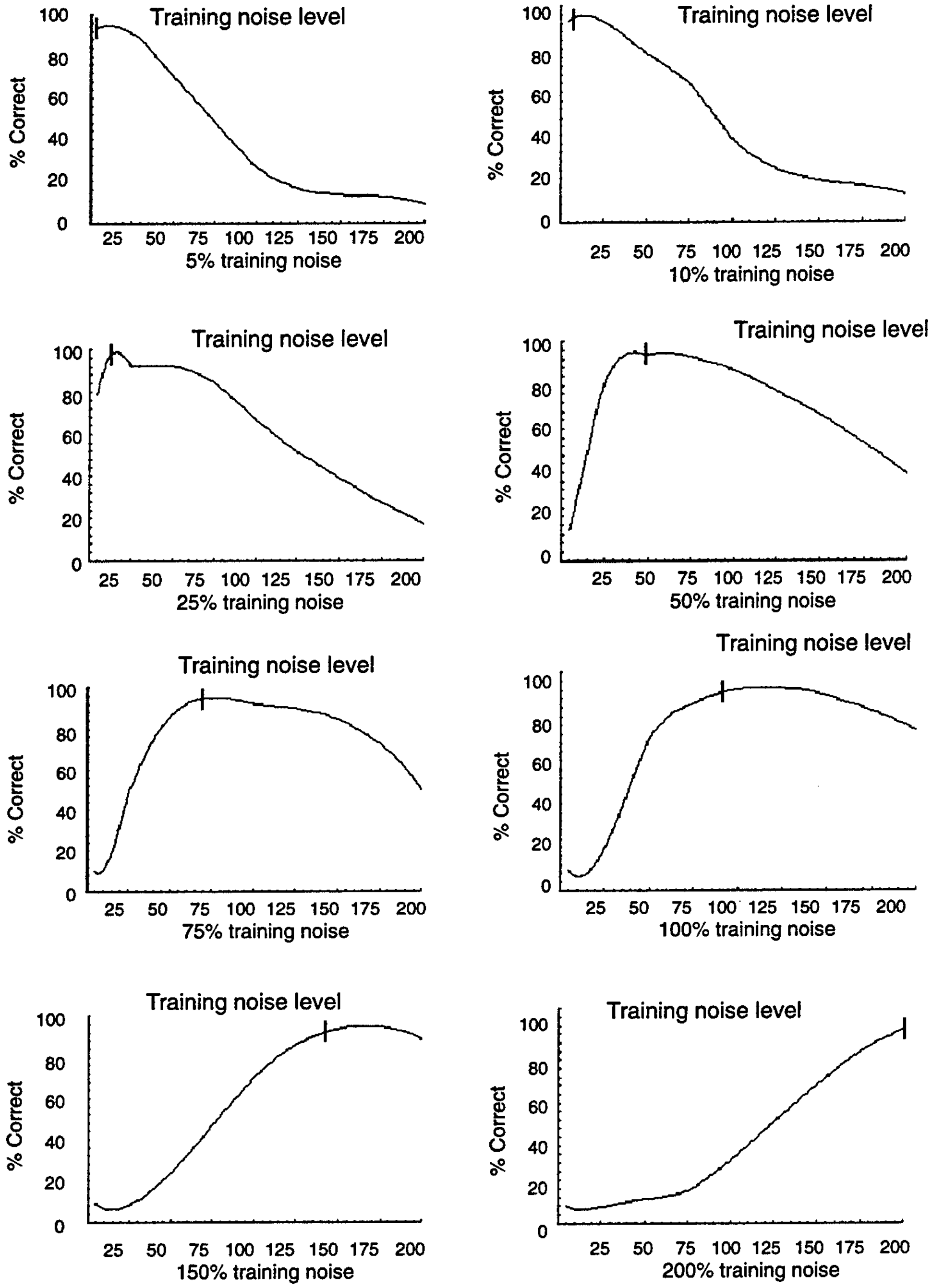
Table 3. Noisy training (639 samples)/noisy testing (71 samples): 1-nearest neighbor.

Training noise level (%)	% correct classification by testing noise percentage							
	5%	10%	25%	50%	75%	100%	150%	200%
5	93.2	94.4	88.9	71.4	52.8	27.8	15.1	8.7
10	93.3	95.2	93.0	77.9	63.9	38.0	18.9	12.5
25	79.7	93.5	92.5	92.4	84.7	66.8	38.9	17.3
50	14.9	31.7	80.7	93.9	93.1	86.8	66.3	39.6
75	9.8	10.3	47.9	85.9	94.8	92.1	84.4	50.4
100	9.2	6.6	18.9	70.3	85.8	92.3	89.3	73.9
150	8.7	6.9	8.3	23.7	46.5	70.1	93.7	89.0
200	7.9	6.5	7.6	11.0	14.8	29.3	64.2	90.1

Table 4. Noisy training (639 samples)/noisy testing (71 samples): 7-nearest neighbor.

Training noise level (%)	% correct classification by testing noise percentage							
	5%	10%	25%	50%	75%	100%	150%	200%
5	93.9	94.7	91.3	70.0	48.5	26.6	13.4	8.9
10	91.1	94.1	91.3	74.8	58.2	35.9	18.6	9.9
25	66.9	86.9	93.5	89.0	84.9	58.6	29.2	15.8
50	14.7	25.8	78.9	93.5	91.6	83.4	60.0	41.1
75	9.4	8.5	37.6	80.1	93.5	91.0	82.1	49.3
100	9.8	7.9	17.0	61.0	78.9	90.7	87.0	71.6
150	8.1	6.9	8.2	21.4	42.8	63.9	92.1	82.3
200	7.8	6.3	7.3	10.4	14.1	29.4	60.4	88.2

Figure 4. Noisy training/noisy testing classification accuracy (1-NN).



6. Analysis of Results

6.1 Experiment 1

The results of experiment 1 show that the *k*-NN approach trained on the clean library spectra is tolerant of low levels of added random noise (the approach gave 93.7-percent correct results in testing with 10-percent added noise); however, performance quickly degrades as higher levels of noise are introduced. Although this performance degradation was not linear, it did not drop off nearly as fast as when the noisy data were used for training and the clean data for testing (see next section).

6.2 Experiment 2

The results of experiment 2 demonstrate that training on noisy data is not effective for classifying the clean library spectra if the noise level exceeds a small amount (5 percent). Performance degradation drops off quickly (nearly a step function) when the added random noise exceeds 5 percent. An interesting observation from experiment 2 (borne out in experiment 3) is that *k*-NN performed increasingly poorly in this domain as more neighbors were considered. This result is unexpected in light of the fact that *k*-NN approaches Bayesian classification accuracy as *k* increases [1]. After examining the data, we determined that the successive degradation in performance as *k* increased was generally due to the misclassification of more instances of the same chemical agent. For example, at a noise level of 5 percent, 1-NN and 3-NN made the same three mistakes: one instance each of agents 71, 57, and 60 was misclassified. 7-NN correctly classified the instance of agent 57, but failed to correctly classify an instance of agent 24 and two additional instances of agent 60. 17-NN misclassified an additional instance of agent 24, as well as two new instances of agent 57 (17-NN did, however, correctly classify the two instances of agent 60 misclassified by 7-NN).

This behavior seems to suggest that the accuracy of *k*-NN for larger values of *k* depends on the differences between positive training instances of the same category. All the instances used for training in this data set carried essentially the same information. They were merely copies of one another with randomly generated differences. Since the amount of additional information available in successive copies is minimal and accidental, the benefit of retaining those additional copies as training instances was outweighed by the fact that the random variations in some of the copies resulted in degenerate, "outlier" instances that are closer to outliers of another class than to their own (fig. 5). It is not the number of positive instances of a particular class that is important for nearest-neighbor classification. Rather, it is how representative each positive instance is of the class. Additional positive instances are potentially detrimental unless they represent different characteristics or aspects of the class not contained in other positive instances. This result suggests that the training data used in these tests should be subjected to some form of editing.

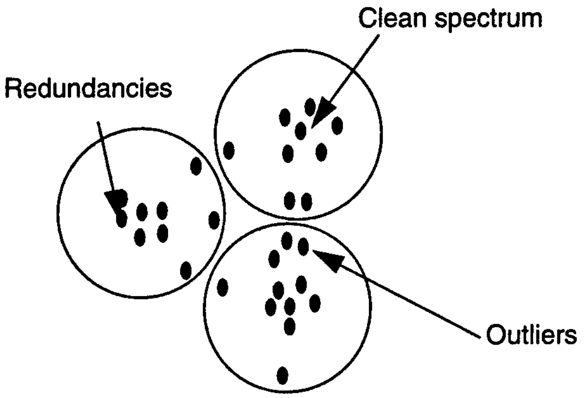


Figure 5. Pictoral explanation of *k*-NN phenomenon observed in experiments 2 and 3.

One form of editing, suggested by Wilson [1], is to eliminate any training instances that are not classified correctly (as determined by the *k*-NN on the remaining instances). This procedure would result in degenerate outliers being eliminated from the training set. Alternatively, one could eliminate all but one of the training instances that are classified correctly. This procedure could result in redundant training instances being eliminated. Yet another approach, suggested by Cost and Salzberg [12], is to leave the degenerate or redundant instances in the training set but define "exception spaces" around them by weighting the conditional probabilities of their feature space[*] according to the number of other exemplars closest to them but in different categories. This approach would result in degenerate outliers having less attractive power (i.e., it limits their "sphere of influence") and effectively distances them further from other "confusables" close to them in the feature space but with different classification categories. Regardless of the approach, some form of editing should be used, since the results of this experiment suggest that indiscriminate inclusion of degenerate or redundant instances in the training set can result in decreased classification accuracy.

6.3 Experiment 3

In experiment 3, every noisy test data set was tested against every noisy training data set. The idea behind this experiment was to see if any particular noise levels were better or worse for training. Although we cannot say that any of the training sets proved especially accurate relative to the others,[†] it is interesting not only that relatively high levels of accuracy

[*] *Cost and Salzberg use a distance metric known as "value difference metric" (VDM), based on the conditional probability (using a relative frequency interpretation) of a specific category given a value for a dimension in the feature space (i.e., p(category | feature)).*

[†] *We performed an F test to test the null hypothesis that the classification accuracy was not affected by the differences in noise levels in the experiments in which the same noise level was used for training and testing (the diagonal of the matrix in table 3). We accomplished this by comparing the sum of squares between the groups (S_b) to the sum of squares within the groups (S_w), where the group is defined as the percentage of noise added before training/testing. Let Y_{ij} represent the i^{th} observation from the j^{th} group:*

$$S_w = \sum_i \sum_j \left(Y_{ij} - \bar{Y}_i\right)^2 ,$$

were obtained on the test data sets that contained the same amount of noise as the training set (which is to be expected), but also that the noisy training data sets seemed to be tolerant of a wider range of noise (and their performance degradation closer to linear) than the clean library spectra training set (fig. 5). This suggests that it may be worthwhile to determine a ballpark estimate of the amount of noise that will be encountered in the field and train on instances with that amount of added noise. Even if the ballpark estimate is not accurate, if it is relatively close, classification accuracy might be increased.

7. Conclusions

The experiments performed here suggest that straightforward nearest-neighbor approaches without editing are tolerant of random noise when the amount of noise in the training set is relatively close to the amount of noise in the test set. Experiment 3, in which the classification system was both trained and tested on noisy data, demonstrated better performance with greater tolerance of different levels of noise than the results obtained by Rauss [13]. Again however, Rauss compared the performance of testing on noisy data only after training on clean, "noise-free" data (i.e., experiment 2 here). We conclude that careful consideration of the training set relative to the testing environment can result in substantially improved performance of nearest-neighbor classification techniques. The best results are obtained when the training and testing sets have roughly the same levels of noise. When the amount of noise in the training set is drastically different from that in the test set, performance degradation was observed.

Finally, the poor performance of *k*-NN compared to 1-NN indicates that additional copies of essentially the same positive instance can degrade classification accuracy. Some form of editing should be done to reduce the effect of degenerate outliers and redundant data.

and

$$S_b = \sum_i n_i \times (\bar{Y}_i - \bar{Y})^2 \ .$$

Based on the 1-NN classification results, S_w *= 351.99 with 72 degrees of freedom (80 tests – 8 groups – 1), and* S_b *= 91.95 with 7 degrees of freedom (8 groups – 1). The F ratio,*

$$\frac{S_b}{S_w} \times \frac{72}{7} = 2.6893 \ ,$$

is compared to the 95-percent confidence level of F(7, 72) = 2.13966. Since the F ratio is less than the 95-percent confidence level of F(7, 72), we reject the null hypothesis that classification is independent of the training/testing noise level with 95-percent confidence.

8. Further Research

Further experiments need to be run with an edited nearest-neighbor technique. Suggested further experiments include (1) editing the training set by throwing out degenerate outliers that are misclassified by the other positive instances of the class, (2) editing the training set by throwing out redundant positive instances of the class, and (3) identifying degenerate positive instances that misclassify other positive instances and weighting them proportionally to their misclassification accuracy, so as to distance them further from other training instances.

We also want to examine and compare other classification techniques on this same problem domain and data sets. Other methods warranting consideration are connectionist-based approaches (neural networks), axis-parallel [4] and oblique decision trees [15], and genetic algorithms [16].

Additionally, a problem of great interest in this domain is the classification of multiple chemical agents present in the same spectrum. This is the example of "clutter" mentioned earlier, where a mixture of agents in the same spectrum causes an interference problem in the identification of each individual agent. To test the classification system's performance on this mixture problem, one would generate test data sets by randomly selecting two different agents and adding their spectral contents. The training sets would remain as presented. With 2-NN or greater used, the system would be penalized most heavily when neither of the agents appeared in the *k*-closest list, penalized less if at least one of the agents appeared in the *k*-closest list, and not penalized at all if both the agents were among the *k*-closest.

Finally, a potentially worthwhile approach that fuses connectionist-based and instance-based classification techniques would be to create a separate perceptron for each category. Test instances would then be fed to each perceptron, and the activation levels of the separate perceptrons would constitute the "distance" (actually the "distance" would be defined as 1.0 – activation level) between the category and the test instance. The *k*-NN technique could then be applied to find the closest classifications.

References

1. Dennis Wilson, *Asymptotic Properties of Nearest Neighbor Rules Using Edited Data*, IEEE Trans. Systems **3** (July 1972).

2. Leslie Pack Kaelbling, *A Formal Framework for Learning in Embedded Systems*, Machine Learning Workshop (1989).

3. Dennis Kibler and Pat Langley, *Machine Learning as an Experimental Science*, Proceedings of the Third European Working Session on Learning, Glasgow, Scotland (1988).

4. J. R. Quinlan, *Induction on Decision Trees,* Machine Learning **1** (1986), 81–106.

5. Ryszard Michalski, *A Theory and Methodology of Inductive Learning,* Artificial Intelligence **20** (1983), 111–161.

6. Tom Mitchell, *Generalization as Search,* Artificial Intelligence **18** (1982), 203–226.

7. Dennis Kibler and David Aha, *Learning Representative Exemplars of Concepts: An Initial Case Study,* Proc. Fourth International Workshop on Machine Learning, Irvine, CA, Morgan Kauffman (1987), pp 24–30.

8. Raymond Mooney, Jude Shavlik, Geoffrey Towell, and Alan Gove, *An Experimental Comparison of Symbolic and Connectionist Learning Algorithms,* Proc. Eleventh International Joint Conference on Artificial Intelligence, Detroit, MI, Morgan Kauffman (1989), pp 775–780.

9. David Haussler, *Quantifying Inductive Bias: AI Learning Algorithms and Valiant's Learning Framework,* Artificial Intelligence **36** (1988), 177–222.

10. Marvin Minsky and Seymour Papert, *Perceptrons: An Introduction to Computational Geometry,* MIT Press, Cambridge, MA (1969).

11. Patrick Rauss, *Noise Tolerance of Model-Based Neural ATR,* Proc. Nineteenth Army Science Conference, Orlando, FL (1994).

12. Scott Cost and Steven Salzberg, *A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features,* Machine Learning **10** (1993), 57–78.

13. Patrick Rauss, *Automatic Recognition of Spectral Components Using Neural Networks,* Army Research Laboratory, unpublished (1992).

14. Jerome Friedman, Jean Louis Bentley, and Raphael Ari Finkel, *An Algorithm for Finding Best Matches in Logarithmic Expected Time,* ACM Trans. Mathematical Software **3**, No. 3 (September 1977).

15. Sreerama Murthy, Simon Ksif, Steven Salzberg, and Richard Beigel, *OC1: Randomized Induction of Oblique Decision Trees,* Proc. of the Eleventh National Conference on Artificial Intelligence (AAAI-93) Washington, DC (1993) 322–327.

16. J. H. Holland, *Adaptation in Natural and Artificial Systems,* University of Michigan Press, Ann Arbor, MI (1975).

Distribution

Admnstr
Defns Techl Info Ctr
Attn DTIC-OCP
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

Hdqtrs Dept of the Army
Attn DAMO-FDQ D Schmidt
400 Army Pentagon
Washington DC 20310-0460

NASA Goddard Spc Flight Ctr
Attn M/S 514 R Schweiss
Greenbelt MD 20771

Johns Hopkins Univ
Applied Physics Lab
Attn E Immer
Attn R Semmel
Johns Hopkins Rd
Laurel MD 20723-6099

Univ of Maryland
Dept of Computer Science
Attn P Godfrey
College Park MD 20745

US Army Rsrch Lab
Attn AMSRL-CI-LL Tech Lib (3 copies)
Attn AMSRL-CS-AL-TA Mail & Records Mgmt
Attn AMSRL-CS-AL-TP Techl Pub (3 copies)
Attn AMSRL-IS L Tokarcik
Attn AMSRL-IS P Emmerman
Attn AMSRL-IS R Winkler (4 copies)
Attn AMSRL-IS S Allen
Attn AMSRL-IS S Ho
Attn AMSRL-IS T Gregory (4 copies)
Attn AMSRL-IS U Movva
Attn AMSRL-IS-C LTC M R Kindl
Attn AMSRL-IS-CI D Hillis
Attn AMSRL-IS-CI T Mills
Attn AMSRL-IS-CS J Gantt
Attn AMSRL-IS-ES COL Price
Attn AMSRL-IS-PA M Salonish
Attn AMSRL-SE-RT P Rauss
Adelphi MD 20783-1197

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
January 1997

3. REPORT TYPE AND DATES COVERED
Final, from April 1993 to April 1996

4. TITLE AND SUBTITLE
Instance-Based Classification of Noisy Infrared Spectra

5. FUNDING NUMBERS
PE: 63734A

6. AUTHOR(S)
Robert P. Winkler and Timothy C. Gregory

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
Attn: AMSRL-IS-PA
2800 Powder Mill Road
Adelphi, MD 20783-1197

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-1211

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783-1197

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
AMS code: 633734.T100211
ARL PR: 7FB330

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Successful systems for classification of real-world data must be tolerant of noise—that is, distortions introduced into the system's model of the real-world domain. Most classification systems are trained on a set of exemplars to identify features of each category and then tested on previously unseen instances. In an instance-based classification system using *k*-nearest neighbor (*k*-NN), the training phase is reduced to storing one or more exemplars for each category. During testing, a distance metric is applied to the features of the new instance to determine the *k* closest exemplars. A voting scheme assigns the category of the modal average to the testing instance. Unlike other methods, *k*-NN does not try to distinguish between "relevant" and "irrelevant" features. Nonetheless, *k*-NN has been shown to asymptotically approach optimal Bayesian accuracy.

This report presents the results of applying *k*-NN to the problem of classifying chemical agents from noisy infrared absorption spectra (from a suite of chemical agents used elsewhere in the literature). Straightforward nearest-neighbor approaches without editing appear to be tolerant of random noise when the amounts of noise in the training and testing sets are relatively close. Performance of *k*-NN versus 1-NN approaches can be improved if the training sets are edited so as to exclude degenerate outliers and redundant positive instances.

14. SUBJECT TERMS
Machine learning, instance-based classification

15. NUMBER OF PAGES
21

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UL